DYNAMIC PORT MANAGEMENT

Inventor: Wei Lu
184 Beechwood Lane
Coppell, Texas 75019
Citizenship: People's Republic of China

Junan Duan
2749 Club Ridge Drive
Lewisville, Texas 75067
Citizenship: The Unites States of America

Assignee: NEC USA, Inc
6535 N. State Highway 161
Irving, Texas 75039-2402

HAYNES AND BOONE, L.L.P.
901 Main Street, Suite 3100
Dallas, Texas 75202-3789
(214) 651-5000
Attorney Docket No. 28272.7
D-873605.2

EXPRESS MAIL NO.: EL828063904US DATE OF DEPOSIT: 4-6-01

This paper and fee are being deposited with the U.S. Postal Service Express Mail Post Office to Addressee service under 37 CFR §1.10 on the date indicated above and is addressed to the Commissioner for Patents, Washington, D.C. 20231.

Gayle Conner — Gayle Conner

Name of person mailing paper and fee — Signature of person mailing paper and fee

DYNAMIC PORT MANAGEMENT

BACKGROUND OF THE INVENTION

[0001] The present invention relates generally to computer network connections in a large scale network environment, and more particularly, to a system and method for providing addresses and ports for specific nodes in the computer network using a dynamic port management module.

[0002] There are many types of computer networks, including local area networks, wide area networks, and the Internet. Companies and organizations often use local or wide area networks as their private networks to link individual nodes (e.g., computers) for email communications, remote access, and internal data sharing. Depending on the sizes of the companies, these private networks can be very large. In order to maintain the integrity of the private networks, the nodes therein are often connected through a gateway to an outside network such as the Internet for additional communication purposes.

[0003] Typically, each node will have a unique network address for the private network. The address, however, may not be of the type or format that is commonly used for the another network with which nodes on the private

network may communicate. For example, the private network may use an address format other than Internet Protocol (IP), while IP addresses are required for the Internet. In this example, the address used in the private network may not be used for communications with nodes connected to (or through) the Internet. In this situation, the gateway will have to assign a registered IP address to the node of the private network that is communicating with or through the Internet.

[0004] What is needed is a system and method for allowing the gateway to properly assign an IP address (or other appropriate address) to facilitate the communication between the nodes of disparate networks.

[0005] In addition to properly assigning an IP address to a node in a private network, the gateway must also control the use of ports that are employed in application sessions. What is also needed also is a system and method for network address mapping along with intelligent dynamic port management.

SUMMARY OF THE INVENTION

[0006] A method and computer program is provided for dynamically managing port and network addresses for a first network to facilitate communications with computing nodes of a second network. According to one example of the present invention, a dynamic port management (DPM) driver is installed on a computing node of the first network and a DPM server is installed on a gateway between the two networks. The first network uses a plurality of network addresses of a first type (e.g., a type that is not "registered" with the second network) for its internal uses and has one or more registered network addresses for communicating with computer nodes in the second network.

[0007] When initiating a communication session with the node in the second network, a "setup" process is then established for exchanging information

between the DPM driver and the DPM server in order to reserve a registered network address and, if the first port is replaceable, for dynamically assigning a second port. The reserved registered network address and the dynamically assigned second port may be used for initiating and completing the communication session.

[0008] If the first port is not replaceable, the first port can be used for future communications. The information exchanged between the DPM driver and the DPM server can also indicate a network address and port for the second node that will be communicated by the first node during the communication session.

[0009] In some embodiments, the DPM server uses at least one unregistered network address and a predetermined port for communications between the DPM driver and the DPM server. Also, a look-up table is created and updated indicating a one-to-one relationship between the reserved registered network address associated with either the first port or the second port (if the first port is replaced) and the first unregistered network address associated with the computing node having the installed DPM driver. This look-up table can also be used for identifying the node and the DPM driver while executing the communication session. This identification feature can be used for continuing the communication session (e.g., an acknowledgment or reply) when information is sent from the second node to the first node of the first network.

[0010] In some embodiments, the DPM server has the ability to reconcile two separate communication sessions requesting the use of the same registered network address and the same port when at least one of the ports is not replaceable. In a typical scenario, the communication session that allows the port to be replaceable will be assigned with a new port by the DPM server.

[0011] In some embodiments, the DPM server has the ability to reconcile two separate communication sessions requesting the use of the same registered network address and the same port when neither session deems the port to be replaceable. In this scenario, both communication sessions are distinguished by using the look-up table to indicate the different destination network addresses.

[0012] Therefore, the present invention achieves significant advantages by allowing the DPM driver and DPM server to assign available ports dynamically for one or more communication sessions.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] Fig. 1 illustrates a schematic of a network computing environment.

[0014] Fig. 2 illustrates a sample data packet.

[0015] Fig. 3 illustrates a schematic showing computer architectural layers for an application, its API, and an IP driver.

[0016] Fig. 4 illustrates a network address translation feature performed by a gateway module.

[0017] Fig. 5 illustrates a layer schematic for including a DPM server-driver pair for managing network address mapping and port assignment according to one example of the present invention.

[0018] Fig. 6 illustrates manipulations made to the packets by the DPM driver and the DPM server of Fig. 5 according to one example of the present invention.

[0019] Fig. 7 illustrates a flow diagram showing a process for completing the network address mapping and the network port management according to one example of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENT

[0020] The present invention provides a new and unique method for dynamic network address and network port management. The disclosure below uses various embodiments to illustrate different features of the invention. These embodiments are intended as examples, and are not intended to limit the invention from that described in the claims.

[0021] Referring now to Fig. 1, a network computing environment 10 includes a private network 12 having internally networked computers 14a-14n. The private network 12 is also connected to the Internet 16 via a gateway 18. In the present example, any computing node or computer 14a-14n inside the private network 12 can communicate with each other, or a computer connectable through the Internet 16 such as a computer 20 or a computer of another private network 22. In furtherance of the example, the information exchanged between any two computers is in the form of data packets and uses a mutually acceptable network protocol such as the Internet Protocol ("IP").

[0022] Referring to Fig. 2, a sample data packet 23 includes header information about the source and destination computers in communication. A first section 24 indicates the IP address of the originating/source host/computer, and a second section 25 indicates the IP address of the destination host/computer. Sections 26 and 28 are identifiers for transport layers (e.g., TCP ports) such as a source port 26 and a destination port 28. The packet 23 also contains sections such as the data section 29a and various other sections (e.g., section 29b and 29c) that may not be directly relevant to the present invention. With the information contained in these sections of the data packet 23, the packet can be routed from network to network, and from computer to computer with ease.

[0023] As of today, an IP address is defined by a 32-bit host address represented in dotted decimal notation (e.g. 10.234.34.4). Limited by its own definition of the 32-bit structure, only 4,294,967,296 unique IP addresses are available for the entire Internet, which far exceed the demands from all the computers connected or connectable to the Internet. Therefore, the private network 12 uses a limited number of IP addresses instead of assigning IP addresses for all the computers 14a-14n. The IP addresses for use with the Internet 16 are called "registered" network addresses, and all others for internal use inside of the private network are known as "unregistered" network addresses. The use of unregistered network addresses inherently generates a conflicting problem for communications between two computers that do not belong to the same private network because all the computers in the private network 12 are not individually identified with their own registered IP addresses.

[0024] Consequently, in order for computers 14a-14n inside the private network 12 to access computers or servers outside, registered IP addresses must be used. Conventionally, the gateway 18 performs network address translation (NAT) or network address port translation (NAPT) to identify and distinguish the source and destination of the transmitted packet to/from the computers 14a-14n. In a more generic term, NAT refers to translations of network addresses and related fields in a packet to make it recognizable to a private network and a public network. NAPT is a specific case of NAT in which modifications are made to the packets in the segments/sections containing transport layer identifiers (e.g., TCP/UDP ports) and their related fields.

[0025] Viewing inside of the private network 12, each computer (e.g., 14a) is assigned independently an IP address which is only known to the private network (i.e., the unregistered IP address or the unregistered network address),

therefore communications among the computers inside the private network can be facilitated. Assuming the private network 12 has a set of registered network addresses or registered IP addresses, there is a mapping mechanism available at the location of the gateway to swap the unregistered IP address to one of the registered IP addresses.

[0026] For the sake of further example, it is assumed that a user on computer 14a initiates an FTP session with a server computer situated outside the private network 12. The computer 14a sends a packet that contains a source IP address of 10.5.5.5 and a destination IP address of 200.2.22.222. The destination IP address indicates that the destination is outside of the private network 12. Since the source IP address 10.5.5.5 is unknown outside of the private network, a return packet from the destination computer using the destination IP address 10.5.5.5 will not reach the computer 14a. Therefore, before the initial packet is sent out from the private network 12, the gateway 18 maps or translates the source IP address to one of the registered IP addresses (e.g., 188.88.8.88). This unique relationship between the unregistered IP address and the mapped registered address is stored in the gateway 18 for future use. With the recognizable IP address of 188.88.8.88, a return packet from the outside server will be delivered to the gateway, and the gateway would once again translate the destination IP address to 10.5.5.5 and forward the packet to computer 14a so that the original FTP session can continue.

[0027] Referring now to Fig. 3, for any particular application on a computer using IP addresses and port numbers (or ports in short), there are three architectural communication entities/layers as shown in block 30, the application 31, the specific application interface (API) 32, and the IP driver 34. When the application initiates a session, it asks the operating system (e.g., Socket) for a port number. The assigned port number, along with the IP address associated with

the computer, is sent to the IP driver, which further furnishes each upcoming packet with the IP address and the assigned port number in its header portion.

[0028] Referring to Fig. 4, conventionally, the gateway 18 uses the NAT feature to simply replace the source's unregistered address with a registered IP address. For example, if the computer in a private network, which bears an IP address of IPx, initiates an FTP session to an outside server having an IP address of IP_{out} and a port number 23, the header portion of the packet will look like block 36. As it has been described with regard to Fig. 2, this header section of the packet indicates that the packet is from a computer having a source IP address of IPx and a source port of 123, and that the packet is intended to be routed to a computer with an IP address of IP_{out} and port 23. When a conventional gateway or other NAT management module receives this packet, the source IP address of the packet is changed to a registered IP address, such as IP_1 as shown in block 38. The IP driver then sends the packet out.

[0029] A lookup table (not shown) is also created to indicate that the IP address-port pair IPx:123 has been changed to IP_1:123. Therefore, when a return packet is received by the gateway bearing the destination IP address of IP_1 and port 123, it can be routed correctly to IPx and port 123. It is noticed that the gateway usually does not change the port number. If the port 123 is used by an application session, then this port will not be available to other applications in the private network for a period of time. This hinders the efficiency of the usage of available ports. On the other hand, if an NAPT is done, and an available port is dynamically chosen by the gateway for sending out the packet, when a return packet comes back bearing the dynamically chosen port number, the application may not be able to further the communication. The reason is that certain applications are required to use a particular port, and an alteration of the port may cause a disruption of future communications.

[0030] Referring now to Fig. 5, a gateway 18 integrated with a Dynamic Port Management (DPM) server is situated between an originating computer 14a with applications and the DPM driver, and a destination computer 14b. Although only one originating computer 14a is shown, it is assumed that a DPM driver is provided at each of the computers 14a-14n of the private network 12 (Fig. 1). According to one example of the present invention, an application 42a communicates with its API 42b, and then, communicates with the DPM driver 42c instead of communicating directly with an IP driver 42d. At the gateway 18, the same structure is formed for a gateway application 44a, its API 44b, the DPM server 44c, and the IP driver 44d for the gateway. It is further noted that the arrows shown in Fig. 5 indicate the direction of communications among different layers. Comparing to Fig. 3, it is clear that the DPM server/driver layer controls information exchanged between the API layer and the IP driver, and thus builds intelligence into the communications among all three layers. With this structure, the IP address and port information is not modified at the packet level, but done by using higher level communications between the DPM driver and the DPM server.

[0031] Referring now to Fig. 6 in conjunction with Fig. 5, the DPM driver/server layer (42c/44c) changes the packets as shown by arrow 48 according to one example of the present invention. Continuing with the FTP session example discussed above, when the computer 14a initiates the FTP session, a communication is first made by the API 42b to the DPM driver 42c installed on the computer 14a to obtain a port number for the session. When the DPM driver 42c assigns the port number to the API 42b, it indicates whether the port number is changeable/replaceable or not. For illustration purposes, it is assumed that the port number assigned is 123 and the IP address is IPx for the computer 14a, and the FTP server in the destination computer 20 bears the IP address of IP_{out} and port 23. Relevant header sections 50 of the outgoing data

packet is shown to include information about IP_x:123 pair and IP_{out}:23 pair. A data section 50a follows the header 50 in the packet. While the application layer 42a conveys this information to the IP driver 42d through its API 42b and the DPM driver 42c, before sending other data packets using the header 50, the DPM driver 42c communicates with the DPM server 44c to "setup" the packets by informing the DPM server 44c that the upcoming FTP session should be directed to the outside FTP server 20.

[0032] This "setup" process may use a plurality of packets communicated between the DPM driver 42c and the DPM server 44c. For instance, any given packet 52 will have a header section 52a. In these packets, the source IP address/port will still be IPx/123 as assigned by the DPM driver, however the destination IP address is now an unregistered IP address of the DPM server IPy, and the port is fixed to a predetermined one of the gateway such as a "well-known" port 1080. The information about the true/final destination (e.g., the destination computer 20) is embedded in a data section 52b of the packet which should include at least, in this case, IP_{out}:23 and an indicator about the replaceability of the port number. It is understood that since this destination information and port replaceability is contained in the data section of the packet, not the header section, various methods can be implemented to have both the DPM driver and server to agree on a predetermined mechanism for each of them to extract such information.

[0033] Also during the setup process, after the DPM server 44c has obtained information about the upcoming FTP session, it informs the DPM driver 42c an appropriate port (e.g., 100) and its IP address (e.g., IPy) for altering the IP address and port information for each packet initiated by the application for the FTP session. The DPM driver 42c "misleads" the IP driver 42d to believe that the packets for the FTP session ought to be sent to the gateway using IPy and port

100 as shown in a sample packet 54 for the FTP session. When the packet 54 arrives at the gateway 18, the DPM server 44c can further instruct the IP driver 44d at the gateway to modify the header of the packet to include appropriate source and destination IP addresses and ports. For instance, a simplified version of an outgoing packet after the DPM server's manipulation is shown as referenced by numeral 56. The source IP address is now changed to a registered IP address (IP_1), the port is changed to 345 (if the port 123 is replaceable), the destination IP address is IP_{out}, and the destination port switches to port 23.

[0034] Referring now to Fig. 7, a flow diagram 70 summarizes the steps taken by the DPM driver and DPM server for manipulating the IP address and port for an application session according to one embodiment of the present invention. Before all the steps are taken, it is assumed that each computer or server is loaded with DPM driver software and the gateway is equipped with DPM server software. Execution begins at step 72, where an application session (communication) is initiated from the DPM driver, and an initial source port is obtained. At step 74, a setup process is executed between the DPM driver and DPM server to inform the DPM server about the final destination IP address and its corresponding port. At step 76, the DPM server checks to see whether the port number may be changed. If so, execution proceeds to step 78 where the DPM server selects an available port and a registered IP address, and dynamically assigns this port for all outgoing packets in the session. Once the port is established (either at steps 76 or 78), execution proceeds to step 80 where the DPM server finds an available registered IP address for outgoing packets. At step 82, a look-up table, which may be stored in the gateway 18, is then updated to reflect the one-to-one relationship between the pair of the originating source IP address (e.g., an unregistered IP address) and the initial port of the application session and the pair of the outgoing registered IP address assigned by the gateway 18 and the dynamically assigned port.

[0035] With the above-described DPM driver-server arrangement and their NAT/NAPT features, any available ports can be dynamically assigned, and thus the efficiency of the gateway is significantly improved. To this end, the DPM server needs to productively manage the availability of the ports to the extent possible. If two application sessions (e.g., two FTP sessions from two different computers) are requesting the same port for their respective sessions, in the conventional method, the gateway can only supply the requested port to one of them, and block the other from using the same port. In the present embodiment, this port "crowdiness" can be resolved by the intelligence of the DPM server-driver.

[0036] For example, consider that a first FTP session is from address IPx and initial port 123, and a second one is from address IPz and port 123. Further, both sessions request the use of a registered address of IP_2. The first session has a destination FTP server at address IP_1 and port 23, and the second session has a destination FTP server at address IP_3 and port 23. In this case, the DPM server will still allow the same registered IP address IP_2 and port 123 to be used concurrently by the two FTP sessions. The reason that the DPM server can do so is because the look-up table can distinguish the two sessions by two different destination IP addresses (i.e., IP_1 and IP_3) for future communications. When a return packet comes back from one of the destination FTP servers, although it is targeted for IP_2 and port 123, it can be identified and routed appropriately base on the fact that the IP addresses of the FTP servers can be differentiated, and that the look-up table provides the unique unregistered IP addresses of the computer inside the private network for further packet routing. If at least one of the initial ports (e.g., 123) can be modified, a new port can be dynamically assigned to replace the initial port. From the perspective of the look-up table, the one-to-one relation between the DPM driver and server can more easily be identified since there is at least one more "differentiator" (i.e., the port used by DPM server for

outgoing packets) available as compared to the situation where neither one of the ports are changeable.

[0037] In the above-described examples, communications between the various computers are discussed. It is well known that a typical computer may include a central processing unit and memory for processing and storing data and programs. The computers may also include external interface devices, such as a modem or network card. It is understood that each of the computers and networks discussed above may be similarly configured, or may be very different. It is also understood that other network nodes, such as mobile nodes using mobileIP, can benefit from the present invention.

[0038] The present disclosure uses the DPM driver-server pair for intelligently and dynamically arranging the use of both the registered IP addresses and the ports for communications among computing nodes to and from a private network. It is understood that the private network is not necessarily limited to a physical location, and the gateway installed with the DPM server is not necessarily located at the same location as the private network. In today's web centric networking environment, a private network can easily exist in a virtual manner in that all the computers/servers belonging to the private network can locate at different locations while still connected to the gateway through the web as long as the gateway can be identified at any moment. To the extent that the gateway is connectable to and accessible by the individual computers, the NAT and NAPT features as described above executed by the DPM server-driver can be carried out seamlessly regardless where the gateway or the computers in the private network are located. It is therefore also contemplated by the present invention that the function of the gateway can be centrally located and provided as an Application Service Provider. This can

reduce the burden of each private network to have its gateway managed independently.

[0039] Another advantage of the present invention is that two different communication components can be used: the DPM driver and the DPM server which adds intelligence on packet processing. Moreover, both the DPM driver and server work together in a symmetric mode of communication. That is, the driver and server work in both communication directions.

[0040] While the invention has been particularly shown and described with reference to the preferred embodiment thereof, it will be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the invention.